A METHOD FOR SPECIFYING NON-TEMPORAL PULSE CHARACTERISTICS

Inventors: Ivan A. Cowie
Mark D. Roberts
Marcus H. Pendergrass
Larry W. Fullerton
James L. Richards

Cross-Reference to Other Applications

The following applications of common assignee may contain common disclosure with the present application:

U.S. Patent Application entitled *"A METHOD AND APPARATUS FOR APPLYING CODES HAVING PRE-DEFINED PROPERTIES"*, filed concurrently herewith, *Attorney Docket No. 28549-160072*.

U.S. Patent Application entitled *"A METHOD AND APPARATUS FOR POSITIONING PULSES USING A LAYOUT HAVING NON-ALLOWABLE REGIONS"*, filed concurrently herewith, *Attorney Docket No. 28549-160073*.

U.S. Patent Application entitled *"A METHOD AND APPARATUS FOR POSITIONING PULSES IN TIME"*, filed concurrently herewith, *Attorney Docket No. 28549-160074*.

U.S. Patent Application entitled *"A METHOD AND APPARATUS FOR MAPPING PULSES TO A NON-FIXED LAYOUT"*, filed concurrently herewith, *Attorney Docket No. 28549-160075*.

U.S. Patent Application entitled *"A METHOD AND APPARATUS FOR SPECIFYING PULSE CHARACTERISTICS USING A CODE THAT SATISFIES PREDEFINED CRITERIA"*, filed concurrently herewith, *Attorney Docket No. 28549-160076*.

U.S. Patent Application entitled *"METHOD FOR SPECIFYING PULSE CHARACTERISTICS USING CODES"*, filed concurrently herewith, *Attorney Docket No. 28549-163655*.

U.S. Patent Application entitled *"A METHOD FOR SPECIFYING NON-ALLOWABLE PULSE CHARACTERISTICS"*, filed concurrently herewith, *Attorney Docket No. 28549-163656*.

The above-listed applications are incorporated herein by reference in their entireties.

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

The present invention relates to impulse transmission systems and, more particularly, to a method of applying codes for specifying non-temporal characteristics of pulses.

RELATED ART

As the availability of communication bandwidth in the increasingly crowded frequency spectrum is becoming a scarce and valuable commodity, Time Modulated Ultra Wideband (TM-UWB) technology provides an excellent alternative for offering significant communication bandwidth, particularly, for various wireless communications applications. Because TM-UWB communication systems are based on communicating extremely short-duration pulses (e.g., pico-seconds in duration), such systems are also known as impulse radio systems. Impulse radio systems were first described in a series of patents, including U.S. Patent Nos. 4,641,317 (issued February 3, 1987), 4,813,057 (issued March 14, 1989), 4,979,186 (issued December 18, 1990), and 5,363,057 (issued November 8, 1994) to Larry W. Fullerton, and U.S. Patent Nos. 5,677,927 (issued October 14, 1997), 5,687,169 (issued November 11, 1997), and 5,832,035 (issued November 3, 1998) to Larry W. Fullerton, et al. These patents are incorporated herein by reference.

Multiple access impulse radio systems are radically different from conventional Code Division Multiple Access (CDMA), Time Division Multiple Access (TDMA) and Frequency Division Multiple Access (FDMA) systems. Unlike such systems, which use continuous sinusoidal waveforms for transmitting information, a conventional impulse radio transmitter emits a low power electromagnetic train of short pulses, which are shaped to approach a Gaussian monocycle. As a result, the impulse radio transmitter uses very little power to generate noise-like communication signals for use in multiple-access communications, radar and positioning applications, among other things. In the multi-access communication applications, the impulse radio systems depend, in part, on processing gain to achieve rejection of unwanted signals. Because of the extremely high achievable processing gains, the impulse radio systems are relatively immune to unwanted signals and interference, which limit the performance of systems that use continuous sinusoidal waveforms. The high

–2–

processing gains of the impulse radio systems also provide much higher dynamic ranges than those commonly achieved by the processing gains of other known spread-spectrum systems.

Impulse radio communication systems transmit and receive the pulses at precisely controlled time intervals, in accordance with a time-hopping code. As such, the time-hopping
5 code defines a communication channel that can be considered as a unidirectional data path for communicating information at high speed. In order to communicate the information over such channels, typical impulse radio transmitters use position modulation, which is a form of time modulation, to position the pulses in time, based on instantaneous samples of a modulating information signal. The modulating information signal may for example be a
10 multi-state information signal, such as a binary signal. Under this arrangement, a modulator varies relative positions of a plurality of pulses on a pulse-by-pulse basis, in accordance with the modulating information signal and a specific time-hopping code that defines the communication channel.

In applications where the modulating information signal is a binary information
15 signal, each binary state may modulate the time position of more than one pulse to generate a modulated, coded timing signal that comprises a train of identically shaped pulses that represent a single data bit. The impulse transmitter applies the generated pulses to a specified transmission medium, via a coupler, such as an antenna, which electromagnetically radiates the pulses for reception by an impulse radio receiver. The impulse radio receiver typically
20 includes a single direct conversion stage. Using a correlator, the conversion stage coherently converts the received pulses to a baseband signal, based on *a priori* knowledge of the time-hopping code. Because of the correlation properties of the selected time-hopping codes, the correlator integrates the desired received pulses coherently, while the undesired noise signals are integrated non-coherently such that by comparing the coherent and non-coherent
25 integration results, the impulse receiver can recover the communicated information.

Conventional spread-spectrum code division multiple access (SS-CDMA) techniques accommodate multiple users by permitting them to use the same frequency bandwidth at the same time. Direct sequence CDMA systems employ pseudo-noise (PN) codewords generated at a transmitter to "spread" the bandwidth occupied by transmitted data beyond the minimum
30 required by the data. The conventional SS-CDMA systems employ a family of orthogonal or quasi-orthogonal spreading codes, with a pilot spreading code sequence synchronized to the family of codes. Each user is assigned one of the spreading codes as a spreading function.

– 3 –

One such spread-spectrum system is described in U.S. Pat. No. 4,901,307 entitled SPREAD-SPECTRUM MULTIPLE ACCESS COMMUNICATION SYSTEM USING SATELLITE OR TERRESTRIAL REPEATERS by Gilhousen et al.

Unlike direct sequence spread-spectrum systems, the time-hopping code for impulse radio communications is not necessary for energy spreading, because the monocycle pulses themselves have an inherently wide bandwidth. Instead, the impulse radio systems use the time-hoping codes for channelization, energy smoothing in the frequency domain, and interference suppression. The time-hoping code defines a relative position of each pulse within a group of pulses, or pulse train, such that the combination of pulse positions defines the communications channel. In order to convey information on such communication channel, each state of a multi-state information signal varies a relative pulse position by a predefined time shift such that a modulated, coded timing signal is generated comprising a train of pulses, each with timing corresponding to the combination of the time position coding and the multi-state modulation.

In one conventional binary approach, pulses are time-modulated forward or backward about a nominal position. More specifically, each pulse is time modulated by adjusting its position within a time frame to one of two or more possible times. For example, in order to send a "0" binary bit during the time frame, the pulse may be offset from a nominal position of the time frame by about -50 pico-seconds. For a "1" binary state, the pulse may be offset from the nominal position by about +50 pico-seconds. Conventional coders that generate the time-hoping code do so in response to a periodic timing signal that corresponds to the data-rate of the multi-state information signal. The data rate of the impulse radio transmission may for example be a fraction of a periodic timing signal that is used as a time base or time reference.

In practice, decoding errors are minimized using distinctive time-hopping codes with suitable autocorrelation and cross-correlation properties. The cross-correlation between any two time-hopping codes should be low for minimal interference between multiple users in a communications system or between multiple target reflections in radar and positioning applications. At the same time, the autocorrelation property of a time-hoping code should be steeply peaked, with small side-lobes. Maximally peaked time-hopping code autocorrelation yields optimal acquisition and synchronization properties for communications, radar and positioning applications.

– 4 –

Various coding schemes with known correlation characteristics are available. For example, algebraic codes, Quadratic Congruential (QC) codes, Hyperbolic Congruential (HC) codes and optical codes have been suggested in the past for coding in impulse radio systems. Generally, based on known assumptions, the coding schemes guarantee a maximum number of pulse coincidences, i.e., hits, for any defined time frame or time frame shift during which the codes are repeated. For example, HC codes are guaranteed a maximum of two hits for any sub-frame or frame shift.

McCorkle in US Patent No. 5,847,677 discloses a random number generator for generating a pseudo-random code for use with jittered pulse repetition interval radar systems. The code is generated by a random number generator that possesses certain attributes desirable for a jittered radar. As disclosed, the attributes related to a flat frequency spectrum, a nearly perfect spike for an autocorrelation function, a controllable absolute minimum and maximum interval, long sequences that do not repeat, and a reasonable average pulse rate.

One known coding technique for an impulse radio is disclosed by *Barrett* in US Patent No. 5,610,907, entitled "Ultrafast Time Hopping CDMA-RF Communications: Code-As-Carrier, Multichannel Operation, High data Rate Operation and Data Rate on Demand." According to the disclosed techniques, two levels of coding are used: major orthogonal codes are applied to provide multiple channels, and forward error correction (FEC) codes are applied to information data before transmission. The disclosed system relies on dividing time into repetitive super-frames, frames and sub-frames. As disclosed, a super-frame corresponds to a time interval of about 1 millisecond, representing one repetition of a code pattern, where as a frame is defined as a time interval of about 1 microsecond divided according to a code length. A sub-frame corresponds to a short time interval of about 1 nano second during which a pulse is time positioned.

Because of practical limitations associated with arbitrarily positioning of pulses in adjacent frames, each frame may have to be divided into allowable and non-allowable time regions for positioning a pulse. One such limitation is associated with hardware limitation on minimum pulse-to-pulse time for respective positioning of two pulses on adjacent frames arbitrarily. The system disclosed in *Barrett* uses a fraction of frame time for encoding and designates the remainder as a RESET period.

Impulse transmission technology may be used in a wide variety of applications, such as multiple-access communication systems, positioning systems, radar systems, etc. To

– 5 –

satisfy the varying requirements of these applications, there exists a need for methods of describing pulse trains more precisely than is now possible using time-hopping codes.

SUMMARY OF THE INVENTION

Time-hopping pulse transmission systems typically transmit pulses having the same non-temporal characteristics (e.g., pulse amplitude) while varying the precise timing of the pulses per a time-hopping code. To improve channelization and spectral response properties of pulse transmission systems and to enable support for multiple applications it is desirable that such systems be capable of transmitting pulses having different non-temporal characteristics. To accommodate such capabilities the present invention provides methods for specifying non-temporal pulse characteristics.

Briefly, according to the present invention, non-temporal pulse characteristics are specified using a code comprising a number of elements. Each code element indicates one or more non-temporal characteristics of a pulse via mapping the code element value to one or more layouts of non-temporal characteristic values. A non-temporal characteristic layout may consist of a range of values such that a code element value maps to a component of the range of values, a sub-component of a component, a smaller sub-component of a sub-component, or even smaller sub-component, ad infinitum. Components, sub-components, and smaller sub-components may contain one or more non-allowable regions. A non-temporal characteristic layout may alternatively consist of a set of discrete values such that a code element value maps to a discrete non-temporal characteristic value.

According to some of the more detailed features of the present invention, the code element may be an integer or a floating-point number. An integer code element may indicate that the value of a pulse characteristic is within a component of a layout comprising a range of values, or alternatively, may indicate a sub-component of a plurality of components. In an exemplary embodiment, the code element is a floating-point number of which the non-fractional part indicates the value of the pulse characteristic is within a component of a layout comprising a range of values, or alternatively, indicates a sub-component of a plurality of components.

Various approaches are used to specify pulse characteristics. In one approach, an established offset value is used to specify an exact characteristic value within the layout component or sub-component specified by the code element value. In an exemplary

– 6 –

embodiment, an absolute ⬤ set value is used to specify the value of ⬤ pulse characteristic by adding the offset value to the minimum characteristic value of a component (or any sub-component of the component) to which a code element value is mapped. In another exemplary embodiment, a relative offset value is used to specify a value of a pulse

5   characteristic that is a fraction of the difference between a minimum characteristic value and a maximum characteristic value of the component (or sub-component of the component, etc.) to which a code element is mapped. In yet another approach, a fractional part of the floating-point number is used to specify a relative offset or fraction of the difference between a minimum characteristic value and a maximum characteristic value of the component (or any

10   sub-component of the component) to which the code element is mapped (using the non-fractional part of the floating-point number). The above approaches are not mutually exclusive.

In another approach, binary representations of code element values are subdivided into multiple groups of bits, where group of bits are translated into decimal values specifying

15   multiple non-temporal characteristics. In another approach the sign of code element values is used to specify a non-temporal characteristic.

The code element can be used to specify a number of non-temporal characteristics of the pulse, including but not being limited to any of: pulse polarity, including whether or not the pulse is inverted; pulse amplitude; pulse width; and pulse type. In one embodiment, the

20   pulse type indicates whether the pulse is a square wave pulse, sawtooth pulse, Haar wavelet pulse, gaussian monopulse, doublet pulse, a triplet pulse, or a set of wavelets. In an exemplary embodiment, code elements can specify temporal pulse characteristics in addition to non-temporal pulse characteristics.

BRIEF DESCRIPTION OF DRAWINGS

The present invention will be described with reference to the accompanying figures, wherein:

Figure 1 illustrates a layout consisting of a pulse amplitude range including exemplary layout parameters, and exemplary subdivisions of the amplitude range including components, sub-components, smaller components, and even smaller components;

Figure 2a illustrates a non-temporal pulse characteristic layout including an exemplary component;

Figure 2b illustrates a non-temporal pulse characteristic layout including multiple components of the same size;

Figure 2c illustrates a non-temporal pulse characteristic layout including multiple components of different sizes;

Figure 2d illustrates a non-temporal pulse characteristic layout including multiple components of the same size that are each subdivided into sub-components of the same size;

Figure 2e illustrates a non-temporal pulse characteristic layout including multiple components of the same size that are each subdivided into sub-components of different sizes;

Figure 2f illustrates a non-temporal pulse characteristic layout including multiple components of different sizes that are each subdivided into sub-components of the same size;

Figure 2g illustrates a non-temporal pulse characteristic layout including multiple components of different sizes that are each subdivided into sub-components of different sizes;

Figure 2h illustrates a non-temporal pulse characteristic layout including multiple same-size components and same-size sub-components combined with smaller components of the same size;

Figure 2i illustrates a non-temporal pulse characteristic layout including multiple same-size components and same-size sub-components combined with smaller components of different sizes;

Figure 2j illustrates a non-temporal pulse characteristic layout including multiple same-size components and different-size sub-components combined with smaller components of the same size;

Figure 2k illustrates a non-temporal pulse characteristic layout including multiple same-size components and different-size sub-components combined with smaller components of different sizes;

Figure 2l illustrates a non-temporal pulse characteristic layout including multiple different-size components and same-size sub-components combined with smaller components of the same size;

Figure 2m illustrates a non-temporal pulse characteristic layout including multiple different-size components and same-size sub-components combined with smaller components of different sizes;

Figure 2n illustrates a non-temporal pulse characteristic layout including multiple different-size components and different-size sub-components combined with smaller components of the same size;

Figure 2o illustrates a non-temporal pulse characteristic layout including multiple different-size components and different-size sub-components are combined with smaller components of different sizes;

Figure 3a illustrates a pulse amplitude value range layout of seven components;

Figure 3b illustrate a pulse width value range layout of five components;

Figure 4a illustrates non-allowable regions within a portion of a non-temporal pulse characteristic layout;

Figure 4b illustrates examples of non-temporal pulse characteristic layouts including non-allowable regions of smaller, same and larger size as compared to allowable regions;

Figure 4c illustrates non-allowable regions within a range of pulse amplitude values, and within layout components, sub-components, and smaller components, with an example of how a code assignment approach can be employed;

Figure 5a illustrates an exemplary discrete pulse amplitude value layout;

Figure 5b illustrates ⬤ exemplary discrete pulse width value layout;

Figure 5c illustrates an exemplary discrete pulse polarity value layout;

Figure 5d illustrates an exemplary discrete pulse type value layout;

Figure 6a illustrates a code mapping approach, depicting integer code element values mapped to a layout encompassing a range of pulse amplitude values subdivided into nine components, where an integer code element exists per pulse, where each code element maps to a component, and where exact pulse amplitude values within components are determined using an offset fraction;

Figure 6b illustrates a code mapping approach, depicting integer code element values mapped to discrete pulse amplitude values, where an integer code element exists per pulse;

Figure 6c illustrates mapping pulse amplitudes to components within a layout encompassing a range of pulse amplitude values using the non-fractional part of a floating point code element value and mapping to exact amplitude values within components using the fractional part of the floating-point code element values;

Figure 6d illustrates mapping of code element values to multiple non-temporal characteristics, where the non-fractional part of each code element specifies pulse amplitude, fractional part of each code element specifies pulse width, and sign of each code element specifies pulse polarity;

Figure 7a illustrates an exemplary embodiment of a least significant bit/most significant bit (LSB/MSB) mapping approach;

Figure 7b illustrates another exemplary embodiment of an LSB/MSB mapping approach;

Figure 8a is a diagram of a linear feedback shift-register pseudorandom number generator; and

Figure 8b is a diagram of an additive Lagged-Fibonacci shift register pseudorandom number generator.

Figure 9 is a block diagram of an impulse transmitter that advantageously uses the present invention.

Figure 10 is a block diagram of an impulse receiver that advantageously uses the present invention.

In the figures, like reference numbers generally indicate identical, functionally similar, and/or structurally similar elements. The figure in which an element first appears is indicated by the leftmost digit(s) in the reference number.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

I.   Overview

In a time-hopping pulse transmission system, a sequence of pulses known as a pulse train is transmitted and received over a period of time. The relative positioning of the pulses in time defines a channel used by the system to transmit information. Time-hopping pulse transmission systems transmit these pulses such that they have the same non-temporal (i.e., non-time dependent) characteristics while varying the precise timing of the pulses per a time-hopping code. (Non-temporal characteristics include amplitude, width, and other non-time dependent characteristics of a pulse or series of pulses.)

However, to improve channelization and spectral response properties of the pulse trains employed by pulse transmission systems and to enable support for multiple applications, it is desirable that such systems be capable of transmitting pulses having different non-temporal characteristics. Accordingly, the present invention provides a method of using a numerically generated code to specify non-temporal characteristics of pulses within a pulse train for use in an impulse transmission system. The method (and accompanying system) includes: defining a non-temporal characteristic layout; identifying non-allowable regions within the layout; specifying an approach for mapping code element values to non-temporal characteristic values; generating a code using a numerical code generation technique; and mapping code element values to non-temporal characteristic values in accordance with the defined layout and code mapping approach.

II.   Value Range Layouts

To facilitate mapping of codes to non-temporal characteristics, a layout of non-temporal characteristic values can be defined. Such layouts can be defined in a multitude of ways to accommodate a wide variety of pulse transmission system applications.

One embodiment involves a value range layout where a range of non-temporal characteristic values is divided into smaller and smaller components to achieve a desired

– 11 –

component resolution in order to facilitate mapping of a code element value to a characteristic value that resides within a component, which corresponds to some range of values. The below described embodiment performs this value range layout using the amplitude of a pulse as the non-temporal characteristic. It should be noted, however, that although much of the below described embodiment pertains to a pulse amplitude characteristic it is generally applicable to other non-temporal characteristics as well. In other words, value range layouts of pulse width, pulse polarity, etc., are also possible. This fact, together with the different types of non-temporal characteristics will be made clear from the below description, together with equivalents thereof, as will be recognized by persons skilled in the relevant art.

Figure 1 illustrates an exemplary value range layout of the pulse amplitude. Element 102 is a value range layout. The pulse takes on amplitude values between a minimum value of a_0 at 110, and a maximum value of a_{max} at 112 in value range layout 102. Value range layout 102 includes components 104. Components 104 include sub-components 106, which in turn include smaller components 108. The process can be repeated, ad infinitum, so that smaller and smaller components can be obtained.

The following describes this process in detail. Referring back to Figure 1, the two layout parameters, a_0 and a_{max}, define the amplitude range, so that it is bounded by minimum pulse amplitude a_0 and maximum pulse amplitude a_{max}. A second layout parameter, $N_{components}$, can be specified to divide the amplitude range 102 into one or more components 104 of the same size, or of different sizes, with each component 104 (indexed by the letter n) having a minimum amplitude value, $a_{min}(n)$, and a maximum amplitude value, $a_{max}(n)$, where $n = 1$ to $N_{components}$.

The number and size of components used in a given layout can be used for various reasons. For example, the number and size of the components can be tailored to meet specific application requirements, to remain within system implementation limits, to achieve one or more of a variety of system characteristics in areas such as performance (i.e., bit rate), reliability (i.e., bit error rate), system-simplicity, ease-of-use, *inter alia*. When different sized components are employed, minimum and maximum amplitude values are specified for each component n, wherein the minimum amplitude value for a given component, $a_{min}(n)$, equals the maximum amplitude value of the preceding component, $a_{max}(n-1)$, or a_0, and the maximum amplitude value of a given component, $a_{max}(n)$, equals the minimum amplitude value for the following component, $a_{min}(n+1)$, or a_{max}. When same sized components are

– 12 –

employed, the amplitude ra●● is evenly divided such that $a_{max}(n)$ - $a_{min}(n)$ is equal for each component 104 indexed by n.

An array of layout parameters, $N_{sub\text{-}components}(N_{components})$, can be specified to subdivide each component 104 into sub-components 106 of the same (or of different sizes) with each sub-component 106 (indexed by m) of the component 104 (indexed by n) having a minimum amplitude value, $a_{min}(n,m)$, and a maximum amplitude value, $a_{max}(n,m)$, where $n = 1$ to $N_{components}$ and $m = 1$ to $N_{sub\text{-}components}(n)$.

As with components, the number and size of sub-components for a given component can also be tailored to meet specific application requirements, to remain within system implementation limits, to achieve one or more of a variety of system characteristics in areas such as performance (i.e., bit rate), reliability (i.e., bit error rate), system-simplicity, ease-of-use, etc., and/or for many other reasons. When different sized sub-components are employed, minimum and maximum amplitude values are specified for each sub-component m of each component n, wherein the minimum amplitude value for a given sub-component, $a_{min}(n,m)$, equals the maximum amplitude value of the preceding sub-component, $a_{max}(n,m-1)$, or the minimum amplitude value of the component in which the sub-component resides, $a_{min}(n)$, and the maximum amplitude value of a given sub-component, $a_{max}(n,m)$, equals the minimum amplitude value for the following sub-component, $a_{min}(n,m+1)$, or the maximum amplitude value of the component in which the sub-component resides, $a_{max}(n)$. When same sized sub-components are employed, components are evenly divided such that $a_{max}(n,m)$ - $a_{min}(n,m)$ is equal for each sub-component m of a component n or for all components such that all sub-components of a given component are of the same size, wherein sub-component sizes may vary from component to component or all sub-components of all components are of the same size depending on the sizes of the components and the numbers of sub-components in the components.

In a manner consistent with the subdivision of components into sub-components, additional multi-dimensional arrays of layout parameters can be used to further subdivide sub-components into smaller components of the same or different sizes, ad infinitum, until a smallest desirable component resolution is attained, with amplitude components at each resolution level having a minimum amplitude value, $a_{min}(n,m,...,a)$, and a maximum amplitude value, $a_{max}(n,m,...,a)$, where $n = 1$ to $N_{components}$, $m = 1$ to $N_{sub\text{-}components}(n)$, ..., and $a = 1$ to $N_{smallest\ components}(n,m,...)$. Such further subdivision of sub-components into smaller and

– 13 –

smaller components enable systems with finer and finer tuning resolution and thus higher and higher fidelity, can increase modulation accuracy, and can be useful for other purposes.

As with components and sub-components, the number and size of these smaller components can also be tailored to meet specific application requirements, to remain within system implementation limits, to achieve one or more of a variety of system characteristics in areas such as performance (i.e., bit rate), reliability (i.e., bit error rate), system-simplicity, ease-of-use, etc., and/or for many other reasons. When different sizes of these smaller components are employed, minimum and maximum amplitude values are specified for each smaller component a, wherein the minimum amplitude value for a component, $a_{min}(n,m,...,a)$, equals the maximum amplitude value of the preceding component, $a_{max}(n,m,...,a-1)$, or the minimum amplitude value of the next higher level component in which the component resides, $a_{min}(n,m,...)$, and the maximum amplitude value of a given component, $a_{max}(n,m,...,a)$, equals the minimum amplitude value for the following component, $a_{min}(n,m,...,a+1)$, or the maximum amplitude value of the next higher level component in which the component resides, $a_{max}(n,m,...)$. When same sized smaller components are employed, the next higher level components are evenly divided such that $a_{max}(n,m,...,a)$ - $a_{min}(n,m,...,a)$ is equal for each component a of a given next higher level component or for all next higher level component such that all components of a given next higher level component are of the same size, wherein component sizes may vary from next higher level component to next higher level component or all components of all higher level components are of the same size depending on the sizes of the next higher level components and the numbers of components within the next higher level components.

At the top of Figure 1, an amplitude range is depicted that is bounded by endpoints of a_0 and a_{max}. Beneath this illustration an equivalent amplitude range is shown that has been subdivided into four components by setting the layout parameter $N_{components}$ to a value of four (4), and the size of each component has been established by setting the minimum and maximum amplitude values of each frame, $a_{min}(n)$ and $a_{max}(n)$, where $n = 1$ to 4. An enlargement of the second component is then shown where the component has been subdivided into twenty sub-components by setting the layout parameter $N_{sub-components}(2)$ to a value of twenty (20), and the size of each sub-component has been established by setting the minimum and maximum amplitude values of the sub-components within component two, $a_{min}(n,m)$ and $a_{max}(n,m)$, where $n = 2$ and $m = 1$ to 20. As illustrated, there are 20 sub-

– 14 –

components 106 in component 104, indexed by $n = 2$, and $m = 1$-20, labeled $a_{min}(2,1)$ through $a_{max}(2,20)$.

An enlargement of the eighth sub-component of component two is then shown where the sub-component has been subdivided into ten smaller components by setting the layout parameter $N_{smaller\_components}(2,8)$ to a value of ten (10), and the size of each smaller component has been established by setting the minimum and maximum amplitude values of the smaller components within sub-component eight of component two, $a_{min}(n,m,l)$ and $a_{max}(n,m,l)$, where $n = 2$, $m = 8$, and $l = 1$ to 10. As illustrated, there are 10 smaller components 108 in sub-component 106, indexed by $n = 2$, and $m = 8$, and $l=1$ to 10, labeled $a_{min}(2,8,1)$ through $a_{max}(2,8,10)$.

It is then shown that these smaller components could be subdivided into x even smaller components (whose size is not shown) using another layout parameter [e.g., $N_{even\_smaller\_components}(2,8,5) = x$], which can be further subdivided ad infinitum. Not shown in the figure are the enlargements of the other components, sub-components, and smaller components, which for this example would also contain twenty sub-components, ten smaller components, and x even smaller components, respectively.

By subdividing an amplitude range into levels of smaller and smaller components and by varying or not varying the size of components at different levels, a multitude of different layout permutations can be defined.

It should be noted, however, that although the present embodiment illustrates the layout of the amplitude range characteristic, many other characteristics can be illustrated instead. For example, the component-based layouts of pulse width, pulse polarity, etc., are also possible, as will be recognized by persons skilled in the relevant art.

Figures 2a through 2o provide examples of different types of layouts that can be established. This group of figures illustrates layout permutations involving nesting of up to three levels of components, which could be subdivided further to additional levels, ad infinitum as will be recognized to persons skilled in the art, to produce additional permutations.

Figure 2a depicts a layout 202 consisting of a single component (not labeled). Figures 2b and 2c present layouts consisting of multiple components of the same size and of different

sizes, respectively. Spe●●ally, in Figure 2a, component 1 204 is ●● same size as component 2 206. In Figure 2c, component 1 208 is of a different size than component 2 210.

Figures 2d and 2e illustrate layouts consisting of multiple components of the same size that are each subdivided into sub-components of the same size or of different sizes, respectively. Specifically, in Figure 2d, component 1 204 is the same size as component 2 206, and the sub-components 212, 214 of component 1 204 are of the same size. In Figure 2e, again component 1 204 is of the same size than component 2 206, but sub-components 216, 218 of component 1 204 are of different sizes.

Similarly, Figures 2f and 2g illustrate layouts consisting of multiple components of different sizes that are each subdivided into sub-components of the same size or of different sizes, respectively. Specifically, in Figure 2f, component 1 208 is of a different same size than component 2 210, and the sub-components 212, 214 of component 1 208 are of the same size. In Figure 2g, again component 1 208 is of a different size than component 2 210, but sub-components 216, 218 of component 1 208 are of different sizes.

Adding another level of component subdivision and varying whether or not component size remains constant produce an additional eight layout permutations, as shown in Figures 2h through 2o.

In Figures 2h and 2i same-size components and same-size sub-components are combined with smaller components of the same size and of different sizes, respectively. At the component level (204, 206) and at the sub-component level (212, 214), Figures 2h and 2i are the same. However, the sub-components of Figure 2h have smaller components 220, 222 that are of the same size, whereas the sub-components of Figure 2i have smaller components 224, 226 that are of different sizes.

In Figures 2j and 2k same-size components and different-size sub-components are combined with smaller components of the same size and of different sizes, respectively. At the component level (204, 206) and at the sub-component level (216, 218), Figures 2j and 2k are the same. However, the sub-components of Figure 2j have smaller components 220, 222 that are of the same size, whereas the sub-components of Figure 2k have smaller components 224, 226 that are of different sizes.

In Figures 2l and 2m different-size components and same-size sub-components are combined with smaller components of the same size and of different sizes, respectively. At

– 16 –

the component level (208, 206) and at the sub-component level (212, 214), Figures 2l and 2m are the same. However, the sub-components of Figure 2l have smaller components 220, 222 that are of the same size, whereas the sub-components of Figure 2m have smaller components 224, 226 that are of different sizes.

In Figures 2n and 2o different-size components and different-size sub-components are combined with smaller components of the same size and of different sizes, respectively. At the component level (208, 206) and at the sub-component level (216, 218), Figures 2n and 2o are the same. However, the sub-components of Figure 2n have smaller components 220, 222 that are of the same size, whereas the sub-components of Figure 2o have smaller components 224, 226 that are of different sizes.

Although a number of variations for the components, sub-components, and smaller components are illustrated, those skilled in the art will recognize that the process of breaking down the components into finer and finer levels of granularity ad infinitum is possible, meaning that the smaller components can be further broken down, and so on, and that any conceivable combination of same and different size components is possible.

Figures 3a and 3b depict the use of value range layouts to specify a range of pulse amplitude values and pulse width values, respectively.

Beginning with Figure 3a, the figure illustrates a range of amplitude values from a_0 110 to a_{max} 112, with the range subdivided into seven components (each labeled 204) as shown vertically on the left side of the figure. Seven pulses 302 are also displayed, which correspond to representative pulses that can be specified using the layout. For the illustrated embodiment, the exact amplitude of each pulse is determined by arbitrarily selecting the approximate middle value of the range component to which a code value (i.e., 1 through 7) would map. The exact values selected within the components can be any other common offset value or a different offset value for each pulse. The layout can also be further subdivided into smaller components to provide greater value resolution, as will be recognized by skilled artisans.

Similarly to Figure 3a, Figure 3b illustrates a pulse width value range 304 between w_0 306 and w_{max} 308 that has been subdivided into five components. The first component 204 and the second component 206 are labeled. Five representative pulses 310 that can be specified using the layout are illustrated, each with a different width value. As with the

– 17 –

amplitude values, the width values used for the five representative pulses 310 are arbitrarily selected as the middle value within each component, and persons skilled in the art will recognize that they could have been selected anywhere within the five components.

III.   Non-Allowable Regions Within Value Range Layouts

5   The present invention permits the establishment of allowable versus non-allowable regions within a layout.   One or more non-allowable regions are regions within a layout within which a characteristic value of a pulse is disallowed. A non-allowable region may be equal to a component, be a subset of a component, or span multiple components.   A non-allowable region can be based solely on particular pulse characteristic values within a

10   component, sub-component, or smaller component or can be based on characteristic values of one or more other pulses.

By default, pulse characteristic values can be set to reside within the components, sub-components or smaller components to which mapped by a code element value.   In this embodiment, a characteristic value cannot be less that the minimum characteristic value or

15   greater than the maximum characteristic value of the component, sub-component or smaller component to which a code element value is mapped.

Non-allowable regions bounded by minimum and maximum characteristic values within components, sub-components, or smaller components can also be defined. Such non-allowable regions can be used to accommodate implementation limitations of the pulse

20   transmission system.

Non-allowable regions can be defined based on characteristic values of other pulses as well.   A minimum characteristic value difference (or characteristic value delta) can be established relative to one or more other pulses.   Similarly, a maximum characteristic value delta can be established relative to one or more other pulses.   Non-allowable regions can also

25   be defined that are bounded by minimum and maximum characteristic value deltas relative to one or more pulses.   Such 'relative' non-allowable regions can also be used to accommodate limitations of the pulse transmission system.   These features are better understood in view of Figures 4a-4c below.   Although these figures are described with respect to amplitudes, skilled artisans will recognize that other pulse characteristics can be described (and used to effect

30   mapping) as well.

– 18 –

Figure 4a depicts ● non-allowable regions within a portion ● of a pulse amplitude value range layout. In the top portion of the figure, a component, sub-component, or smaller component is shown, which is bounded by a minimum amplitude value 402 and a maximum amplitude value 404. The amplitude values of two exemplary pulses labeled Pulse 1 406 and Pulse 2 408 are shown. Non-allowable regions A 410, B 412, C 414, D 416 and E 418 are shown in layers to illustrate that they can overlap one another.

Two non-allowable regions 410 are shown corresponding to rule A, which is used to facilitate mapping a code element value, where one region consists of all values less than the minimum amplitude value of the component to which a code element value is mapped and one region consists of all values greater than the maximum amplitude value of the same component.

Three non-allowable regions 412, denoted B, are shown. Rule B regions are non-allowable regions defined as being bounded by minimum and maximum amplitude values.

Four regions 414 are defined by rule C. Rule C regions are non-allowable regions defined by a minimum value delta (or difference) relative to Pulse 1 amplitude value 406 and Pulse 2 amplitude value 408. The first two regions 414 represent where a value cannot be used because it would violate the minimum value delta relative to the Pulse 1 406 amplitude value, and the other two regions 414 represent where a value cannot be used because it would violate the minimum value delta relative to the Pulse 2 408 amplitude value. Conversely, two regions 416, are defined by rule D, and illustrate the application of a maximum delta (or difference) rule, where the leftmost region 416 represents where a value cannot be used because it would violate the maximum value delta relative to the Pulse 2 408 amplitude value, and the rightmost region 416 represents where a value cannot be used because it would violate the maximum value delta relative to the Pulse 1 406 amplitude value. (Two other regions of the same type as those shown in D are not shown due to the size of the figure. These regions would be placed in the far right and far left of the figure.)

Four non-allowable regions 418 are defined by rule E. The rule E regions are non-allowable regions defined as being bounded by minimum and maximum value deltas relative to the two pulses 406, 408.

The bottom portion of Figure 4a depicts the combined effect of the non-allowable regions. Here, the separated layers in the upper portion of the figure are overlaid to show that a given value within a component is either allowable or non-allowable.

Figure 4b depicts relative size relationships between non-allowable and allowable regions within components using these exemplary embodiments. As shown in the three examples, non-allowable regions can be smaller than, the same size as, or larger than allowable regions. Referring to example 1 of Figure 4b, the non-allowable regions 426 of components 420 and 422 are smaller in size than the allowable regions 424. In example 2, the non-allowable regions 430 are the same size as the allowable regions 428. In example 3, the non-allowable regions 434 are larger in size than the allowable regions 432.

Non-allowable regions can also be defined at more than one component level as is shown in Figure 4c. In the figure a layout is divided into two allowable regions 436, 442 and a non-allowable region 438. The first allowable region 436 is divided into three components 440 and the second allowable region is treated as a single component 442. The figure further illustrates how components 440 can contain non-allowable regions 446, sub-components 444 can contain non-allowable regions, and so on. These regions of Figure 4c are further explained below.

IV.   Discrete Value Layouts

Another approach to defining a layout of non-temporal characteristic values is to specify a layout of discrete non-temporal characteristic values to which individual codes can map. Figures 5a through 5d provide examples of discrete value layouts for pulse amplitude, pulse width, pulse polarity, and pulse type, respectively. In Figure 5a, seven discrete values 502 are shown. Representative pulses 504 are depicted having amplitudes that correspond to the seven discrete values. Unlike the representative pulses depicted in Figure 3a, which can have any amplitude within a given range component, the amplitude values of the seven representative pulses in Figure 5a each must equal a discrete amplitude value. Similarly, Figure 5b shows a layout of five discrete pulse width values 506 and five representative pulses 508 having widths that correspond to those discrete values.

The discrete value layout approach also lends itself to non-temporal characteristics that do not involve magnitude. Unlike the pulse amplitude and pulse width characteristics, where either a value range layout or a discrete value layout can be used, other non-temporal characteristics can only be specified using a discrete value layout (e.g., pulse polarity and

– 20 –

pulse type). Figures 5c and 5d illustrate discrete pulse polarity values and pulse type values, respectively. In Figure 5c a regular pulse 510 and an inverted pulse 512 are shown having the same magnitude, but which are the inverse in polarity to one another. In Figure 5d, eight different pulse types 514-528 are shown which have little relationship to one another. These variations specifically include a square wave pulse 514, a sawtooth pulse 516, a Haar wavelet pulse 518, a Gaussian monopulse 520, a doublet pulse 522, a triplet pulse 524, a doublet/triplet wavelet set 526, and other wavelet sets 528 (including any combination of wavelets). Persons skilled in the art will recognize that any combination of the above-noted pulse types can be used as well.

V. Non-Fixed Layouts

The value range and discrete value layout approaches can be similarly employed using non-fixed layouts, which are also referred to as delta layouts. Where a 'fixed' value range layout defines a range of amplitude values between for example a_0 110 and a_{max} 112 (see Figure 1), a 'non-fixed' delta value range layout can specify a range of delta values between for example Δa_0 and Δa_{max}, where the delta value range can be divided into components and sub-components in the same manner as a fixed value range can be divided.

Similarly, where discrete values are defined in a discrete value layout, discrete delta values can be defined in a discrete delta value layout. These non-fixed layout approaches to specifying non-temporal pulse characteristics depend on the non-temporal characteristics of the previous pulse. Specifically, the value of a non-temporal characteristic value of a given pulse equals the non-temporal characteristic value of the previous pulse plus a delta characteristic value. This delta value is determined by mapping a code element value to a defined delta value range layout, or to a discrete delta value layout, as appropriate to a person skilled in the art.

VI. Code Mapping Approaches

A multitude of code mapping approaches can be employed in conjunction with a given layout to accommodate pulse transmission system implementation requirements and to optimize system performance for specific applications. A code specifying non-temporal characteristics can be comprised of elements having either integer or floating-point numbers that map to characteristic values within defined characteristic layouts. The value of each integer code element or the non-fractional part of each floating-point code element value can identify a component, sub-component or smaller component (or even smaller component,

– 21 –

Venable Ref: 28549-165559

calculated ad infinitum) corresponding to a range of characteristic values within a defined value range layout or can identify a discrete value within a defined discrete value layout. Additionally, an integer or floating-point code element can be used to specify multiple non-temporal characteristics.

5   The numbering scheme used for a code must be consistent with the numbering scheme used for the defined characteristic layout to allow mapping of code element values to values within the layout. Components within a value range layout or values within a discrete value layout can be numbered beginning with zero, one, or any other number. Components produced by subdividing higher level components, such as sub-components or smaller

10   components, can be numbered per higher level component such that sub-component numbering begins again with each higher level component, or alternatively can be numbered in sequence independent of the higher level component in which they reside. A non-allowable region can be numbered or not numbered depending on whether code elements are to be mapped to the region.

15   Figure 4c illustrates how different component numbering approaches can be employed. At the top of the figure, the non-allowable region 438 within the amplitude range is effectively a fifth non-numbered component to which a code element cannot be mapped. Similarly, the first component 440 is shown containing an allowable region subdivided into eight sub-components 444 and a non-allowable region 446 numbered as the ninth sub-

20   component, thereby enabling a code element to be mapped to it. The figure further illustrates how a non-allowable region can span multiple components. The first non-allowable region 460 within the sixth sub-component of the first component 440 spans two non-numbered smaller components.

      Once an integer value or non-fractional part of a floating-point value has been used to

25   map to a value range corresponding to a component, sub-component, or smaller component, an established offset value can be used to specify the exact characteristic value within the component, sub-component, or smaller component. Here, a value is acceptable if it resides outside non-allowable regions defined by established rules. An absolute offset value can be used to specify an exact characteristic value that is a common value difference (or delta) from

30   the minimum characteristic values of the components, sub-components, or smaller components to which code elements are mapped. Alternatively, a relative position offset value can be used to specify an exact characteristic value that is a fraction of the difference

– 22 –

between the minimum and maximum characteristic values of the components, sub-components, or smaller components to which code elements are mapped.

For floating-point codes, the fractional part of each floating-point code element can be used to specify the relative offset used to specify an exact characteristic value that is a fraction of the difference between the minimum and maximum characteristic values of the component, sub-component, or smaller component to which the non-fractional part of the floating-point code element is mapped.

Unlike integer codes, where an absolute or relative offset value can be used to specify exact characteristic values in a common manner, the fractional portions of floating-point values can vary per code element. This allows each pulse characteristic value to be established independent of other pulse characteristic values.

Figures 6a through Figures 6d illustrate these component numbering and code mapping approaches. Figure 6a depicts code elements 602 of a code 606 mapped to components 604 within an amplitude range value layout 600, where an integer code element 602 exists per pulse 608. In the illustrated embodiment, a common relative offset fraction of .5 is used to determine the exact amplitude values.

Figure 6b depicts code elements 602 of a code 606 mapped to discrete amplitude values 610 within a discrete value layout 601, where an integer code element 602 exists per pulse 608. It should be noted that the two figures present two different methods for mapping codes to the same amplitude values.

Figures 6c and 6d depict two methods of mapping pulses to frames using floating-point values. In Figure 6c, the non-fractional portion 612 of each floating-point code element value 616 maps a pulse 608 characteristic to a component 604 within an amplitude value range layout 600. The fractional portion 614 of each floating-point code element 616 specifies the fraction of the difference between the minimum 618 and maximum 620 amplitude values of each mapped component 604 used to determine the exact characteristic value for a pulse.

In Figure 6d, the non-fractional part 612 of each floating-point code element value 616 maps to a discrete amplitude 610 per a discrete value layout 601. The fractional part 614 of the floating-point code element value 616 maps to a discrete pulse width 624 per a discrete value layout (not shown). The sign 622 of each floating-point code element value 616 maps

to a discrete pulse polarity 726 (i.e., whether the pulse is inverted) per a discrete value layout (also not shown). The code mapping approaches shown in Figures 6a and 6c can be applied at any component level defined within a value range layout.

A Least Significant Bit/Most Significant Bit (LSB/MSB) approach can also be used to map code elements to characteristic values within a value range or discrete value layout. In one exemplary embodiment of this approach, a component and an offset value can be specified by subdividing the binary representation of a code element value into its most significant and least significant bits.

Here, one subset of bits specifies a component within an amplitude value range layout and the other subset of bits specifies an absolute or relative position offset value used to determine the exact characteristic value within the specified component.

An example of a LSB/MSB mapping approach is shown in Figure 7a. In Figure 7a, the decimal value 702 of each code element value is represented as a 16-bit binary word 704 that is subdivided into its eight MSBs 706 and its eight LSBs 708. The binary value of the MSBs 706 is translated into a decimal value 710 representing the component corresponding to a range of pulse amplitude values. The binary value of the LSBs 708 is translated into a decimal value that is then divided by 100 in order to produce a value 712 representing the desired offset fraction used to calculate the exact amplitude value within the component specified by the MSBs. The bits of a code element's binary representation can also be subdivided into any number of subsets to enable encoding of additional characteristic values (i.e., pulse width, pulse polarity, pulse type, and/or pulse time position).

Figure 7b is similar to Figure 7a except it demonstrates a slightly more complex bit-group mapping approach where the first seven LSBs 710 are used to specify the offset fraction 714 and the last LSB 716 is used to specify pulse polarity 718. The translated bit patterns are shown combined into a translated code that is nearly the same as the translated code shown in Figure 7a, except the first and fifth code element values have a negative sign to specify that the first and fifth pulses are to be inverted.

The LSB/MSB or more complex bit-group mapping method can be employed using binary word sizes that are 8-bit, 16-bit, 32-bit or whatever size is deemed appropriate to encode data describing a pulse train. Given a word size, bits can be grouped as appropriate to encode the largest possible value for a given characteristic. For example, to encode offset

fraction values with 2-digit precision, a group of 7 data bits is required to represent integer values from 0 to 100. Similarly, if 1000 components exist in a layout, a group of 10 bits is required to represent integer values from 1 to 1000. The LSB/MSB or more complex bit-group mapping method can also be employed in combination with established absolute or relative offset values, as will be recognized by persons skilled in the art.

VII.   Numerical Code Generation

After the characteristic layout and code mapping approach have been established, a code is generated using a numerical code generation technique. One such technique involves generation of a code comprised of pseudorandom numbers. A pseudorandom code can be generated using a computer's random number generator, binary shift-register(s) mapped to binary words, a chaotic code generation scheme, or another well-known technique.

Computer random number generator functions commonly employ the linear congruential generation (LCG) method, which generates the n-th random number, x_n, from the previous random number, x_{n-1}, using an equation of the general form:

$$ x_n = Ax_{n-1} + c \left(\mathrm{mod}\ m \right) $$

where n identifies a given code in the generated code sequence, and the generated sequence is characterized by the multiplier A, the additive constant c, the modulus m, and an initial seed x_0. These random number generator functions can be referred to as $LCG(a,c,m,x_0)$, which determines the sequence generated.

Another method used for computer random number generator functions is the Additive Lagged-Fibonacci Generator (ALFG) method. This approach can be described by an equation of the form:

$$ x_n = x_{n-j} + x_{n-k} \left(\mathrm{mod}\ 2^m \right), j < k $$

where n identifies a given code in the generated code sequence, and j and k represent offsets to previously generated codes. The period of these generators is $(2^k-1)2^{m-1}$ and they are referred to as $ALFG(l,k,m,x_0)$, which determines the sequence generated.

– 25 –

Binary shift-register pseudorandom number generators can be implemented in many different ways. One common approach is the linear feedback shift register illustrated in Figure 8a, including exclusive OR gate 802 and registers 804, which can be described by an equation of the form:

$$x_{n+k} = \sum_{i=0}^{k-1} a_i x_{n+i} \, (\text{mod } 2)$$

where n identifies a given code in the generated code sequence, k is the number of bits in the shift register, a_i is the value of the i-th bit in the shift register. The sequence of bits that is generated depends on the initial shift-register state and which shift-register bit value, a_i, is fed back into the exclusive-OR device along with the shifted output.

The ALFG method can also be implemented using a shift register 808 and a modulo adder 806 device, as shown in Figure 8b, which can be described by an equation of the form:

$$x_n = x_{n-j} + x_{n-k} \, (\text{mod } 2), \, j < k$$

where n identifies a given code in the generated code sequence, and j and k represent the shift-register bits fed into the modulo adder device.

Many other different but related traditional approaches for generating pseudorandom code sequences can also be employed, comprising inversive congruential generators, explicit-inversive congruential generators, multiple recursive generators, and combined LCGs. Any of these or other similar traditional methods, as known to skilled artisans, can be used to generate a pseudorandom code without departing from the scope of the invention.

In addition to the more traditional approaches described for generating pseudorandom codes, several other non-traditional approaches can be employed including chaotic code generators and Optimal Golomb Ruler (OGR) code generators. Any of these or other non-traditional methods can also be used to generate a pseudorandom code without departing from the scope of the invention.

Venable Ref: 28549-165559

VIII. Mapping Code Elements to Pulse Characteristic Values

According to the invention, after a code has been generated using one of the above-noted numerical code generation techniques, the code element values are associated with or mapped to specific pulse characteristic values per the defined characteristic layout and the employed code mapping approach. As stated above, non-temporal pulse characteristics include a pulse width characteristic, a pulse amplitude characteristic, a pulse polarity characteristic, and a pulse type characteristic, etc. Of course, code element values may be associated with a non-temporal pulse characteristic in addition to a temporal pulse characteristic, such as a pulse position in time.

A code element value comprises an integer or floating-point value, with each code element value indicating any one of one or more components, one or more sub-components of said component, and one or more smaller components of said sub-component, established by recursively breaking down said sub-component into smaller parts. The one or more components, sub-components, and any number of smaller components may be defined within a layout comprising a range of non-temporal pulse characteristic values. Again, the one or more components are any one of a same size; and a different size than others of said one or more components.

In the manner stated above, any component may include one or more non-allowable regions established by various rules. A rule may, for example, establish non-allowable regions based on a minimum value or a maximum value of a component or may be based on minimum and maximum values within a component. A rule could be based on non-temporal characteristic values of one or more other pulses or a minimum value difference or a maximum value difference from non-temporal characteristic values of one or more other pulses. A rule may also establish a region bounded by a minimum and maximum value difference from non-temporal characteristic values of one or more other pulses.

An established offset value can be used to specify an exact non-temporal characteristic value within a value range layout component specified by a code element value. An absolute offset value could be added to the minimum value of a component. Alternatively a relative offset value may be used to specify a value that is a fraction of the difference between the minimum value and maximum value of a component. The fractional part of a floating-point code element value could be used to specify a relative offset value.

– 27 –

According to one aspect of the present invention, each code element value corresponds to a value that is defined within a layout comprising discrete non-temporal pulse characteristic values. According to another aspect of the present invention, each code element value corresponds to a value that is defined within a layout comprising a range of non-temporal pulse characteristic values and discrete non-temporal pulse characteristic values. As stated above, the present invention can also be employed using a delta value layout.

IX.   Exemplary Transceiver Implementation
Transmitter

An exemplary embodiment of an impulse radio transmitter 902 of an impulse radio communication system having one subcarrier channel will now be described with reference to Fig. 9.

The transmitter 902 comprises a time base 904 that generates a periodic timing signal 907. The time base 904 typically comprises a voltage controlled oscillator (VCO), or the like, having a high timing accuracy and low jitter, on the order of picoseconds (ps). The voltage control to adjust the VCO center frequency is set at calibration to the desired center frequency used to define the transmitter's nominal pulse repetition rate. The periodic timing signal 907 is supplied to a precision timing generator 908.

The precision timing generator 908 supplies synchronizing signals 910 to the code source 912 and utilizes the code source output 914 together with an internally generated subcarrier signal (which is optional) and an information signal 917 to generate a modulated, coded timing signal 918.

The code source 912 comprises a storage device such as a random access memory (RAM), read only memory (ROM), or the like, for storing suitable codes and for outputting the codes as a code signal 914. Alternatively, maximum length shift registers or other computational means can be used to generate the codes.

An information source 920 supplies the information signal 917 to the precision timing generator 908. The information signal 917 can be any type of intelligence, including digital bits representing voice, data, imagery, or the like, analog signals, or complex signals.

A pulse generator 922 uses the modulated, coded timing signal 918 as a trigger to generate output pulses. The output pulses are sent to a transmit antenna 924 via a

– 28 –

transmission line 927 coupled thereto. The output pulses are converted into propagating electromagnetic pulses by the transmit antenna 924. In the present embodiment, the electromagnetic pulses are called the emitted signal, and propagate to an impulse radio receiver 1002, such as shown in Fig. 10, through a propagation medium, such as air, in a radio frequency embodiment. In a preferred embodiment, the emitted signal is wide-band or ultrawide-band, approaching a monocycle pulse as in Figures 3a and 3b. However, the emitted signal can be spectrally modified by filtering of the pulses. This filtering will usually cause each monocycle pulse to have more zero crossings (more cycles) in the time domain. In this case, the impulse radio receiver can use a similar waveform as the template signal in the cross correlator for efficient conversion.

Receiver

An exemplary embodiment of an impulse radio receiver 1002 (hereinafter called the receiver) for the impulse radio communication system is now described with reference to Fig. 10. More specifically, the system illustrated in Fig. 10 is for reception of digital data wherein one or more pulses are transmitted for each data bit.

The receiver 1002 comprises a receive antenna 1004 for receiving a propagated impulse radio signal 1007. A received signal 1008 from the receive antenna 1004 is coupled to a cross correlator or sampler 1010 to produce a baseband output 1012. The cross correlator or sampler 1010 includes multiply and integrate functions together with any necessary filters to optimize signal to noise ratio.

The receiver 1002 also includes a precision timing generator 1014, which receives a periodic timing signal 1017 from a receiver time base 1018. This time base 1018 is adjustable and controllable in time, frequency, or phase, as required by the lock loop in order to lock on the received signal 1008. The precision timing generator 1014 provides synchronizing signals 1020 to the code source 1022 and receives a code control signal 1024 from the code source 1022. The precision timing generator 1014 utilizes the periodic timing signal 1017 and code control signal 1024 to produce a coded timing signal 1027. The template generator 1028 is triggered by this coded timing signal 1027 and produces a train of template signal pulses 1030 ideally having waveforms substantially equivalent to each pulse of the received signal 1008. The code for receiving a given signal is the same code utilized by the originating transmitter 902 to generate the propagated signal 1007. Thus, the timing of the template pulse train 1030 matches the timing of the received signal pulse train 1008,

– 29 –

allowing the received sign̄1008 to be synchronously sampled in the correlator 1010. The correlator 1010 ideally comprises a multiplier followed by a short term integrator to sum the multiplier product over the pulse interval.

Further examples and details of correlation and sampling processes can be found in commonly owned patents 4,641,317, 4,743,906, 4,813,057, and 4979,186, which are incorporated herein by reference, and commonly owned and co-pending application 09/356,384, filed July 16, 1999, titled: "Baseband Signal Converter Device for a Wideband Impulse Radio Receiver," which is incorporated herein by reference.

The output of the correlator 1010, also called a baseband signal 1012, is coupled to a subcarrier demodulator 1032, which demodulates the subcarrier information signal from the subcarrier. The purpose of the optional subcarrier process, when used, is to move the information signal away from DC (zero frequency) to improve immunity to low frequency noise and offsets. The output of the subcarrier demodulator 1032 is then filtered or integrated in a pulse summation stage 1034. The pulse summation stage produces an output representative of the sum of a number of pulse signals comprising a single data bit. The output of the pulse summation stage 1034 is then compared with a nominal zero (or reference) signal output in a detector stage 1038 to determine an output signal 1039 representing an estimate of the original information signal 917.

The baseband signal 1012 is also input to a lowpass filter 1042 (also referred to as lock loop filter 1042). A control loop comprising the lowpass filter 1042, time base 1018, precision timing generator 1014, template generator 1028, and correlator 1010 is used to generate a filtered error signal 1044. The filtered error signal 1044 provides adjustments to the adjustable time base 1018 to time position the periodic timing signal 1027 in relation to the position of the received signal 1008.

In a transceiver embodiment, substantial economy can be achieved by sharing part or all of several of the functions of the transmitter 902 and receiver 1002. Some of these include the time base 1018, precision timing generator 1014, code source 1022, antenna 1004, and the like.

X.   Conclusion

While various embodiments of the present invention have been described above, it should be understood that they have been presented by way of example only, and not

limitation. Thus, the breadth and scope of the present invention should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents.

5